

FIG. 1

```
glMatrixMode( GL_PROJECTION );
glLoadMatrix( intrinsic matrix of projector );
glMultMatrix( xform for rendering view )
glMultMatrix( inverse(xform for shading view) );
glMatrixMode( GL_MODELVIEW );
glLoadMatrix( xform for shading view );
// set virtual light position(s)
// render graphics model
}
```

200

FIG. 2

FIGURE 3D - FIGURE 3E FIGURE

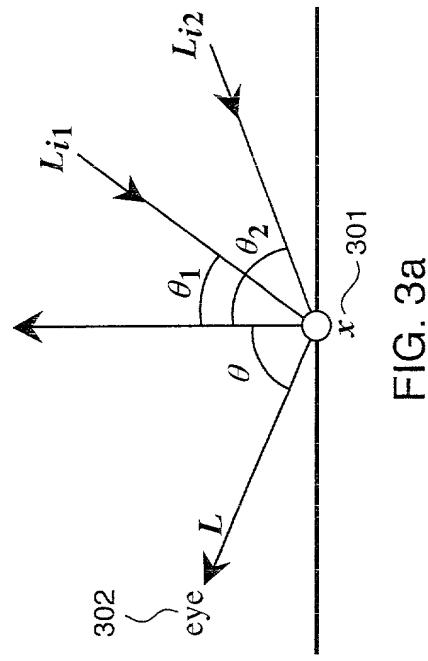


FIG. 3a

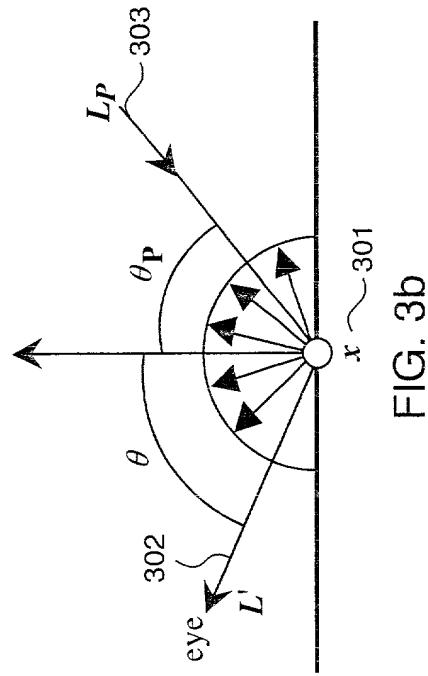


FIG. 3b

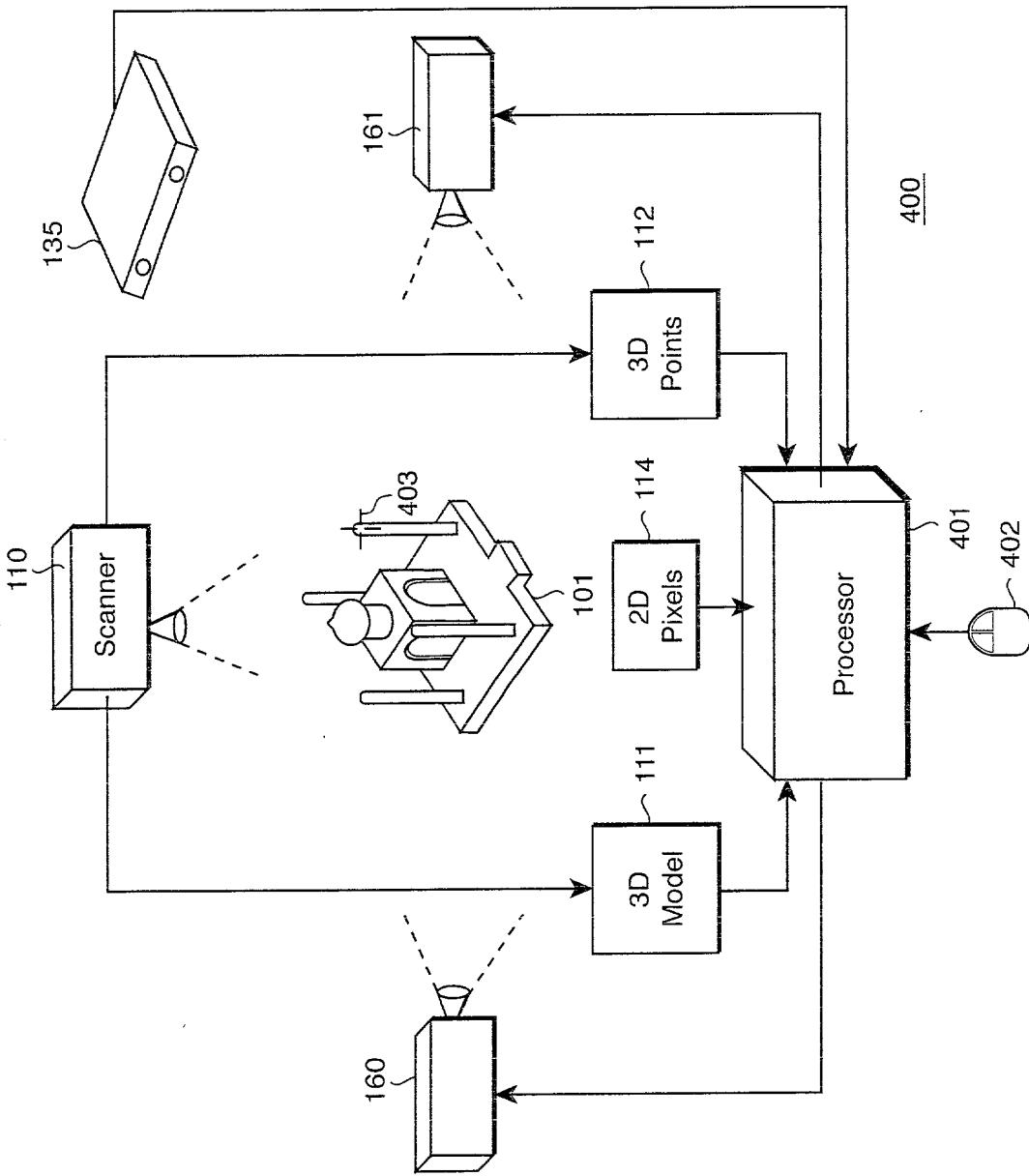


FIG. 4

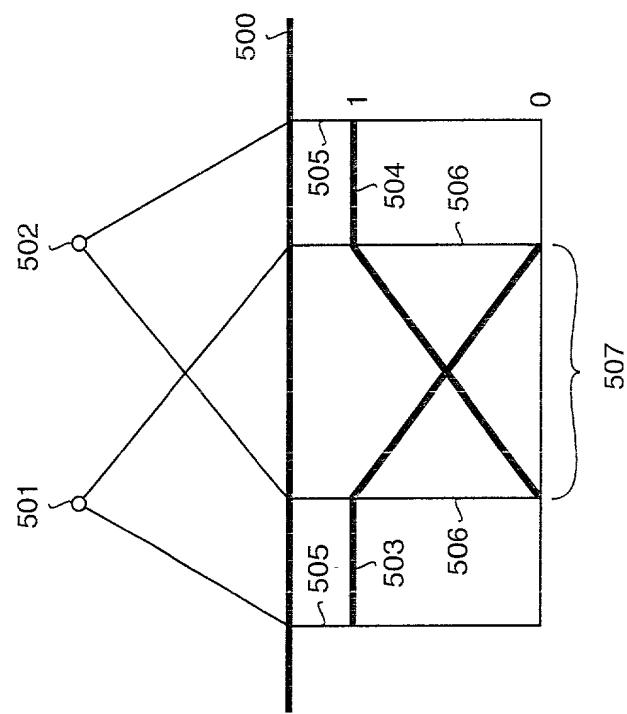


FIG. 5  
PRIOR ART

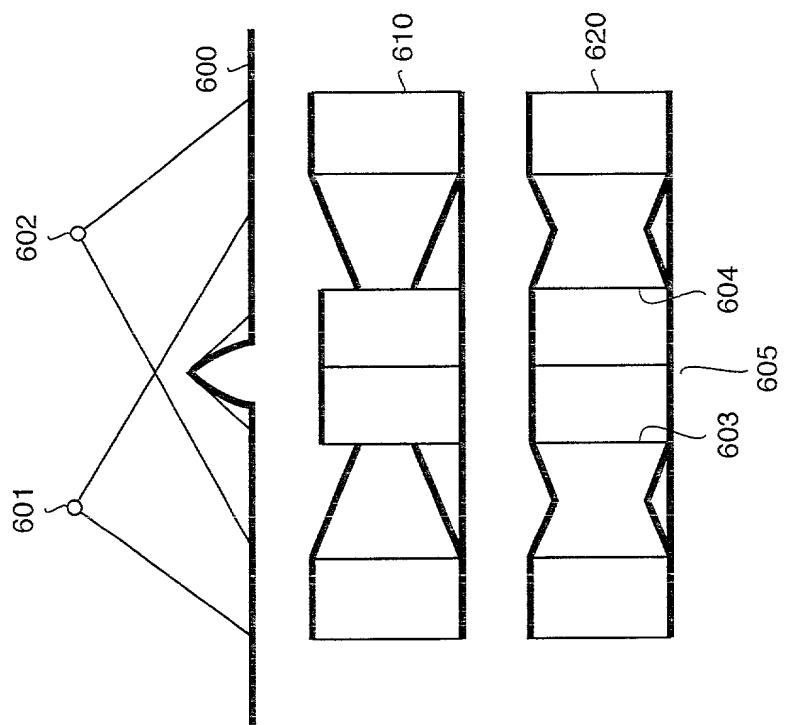


FIG. 6

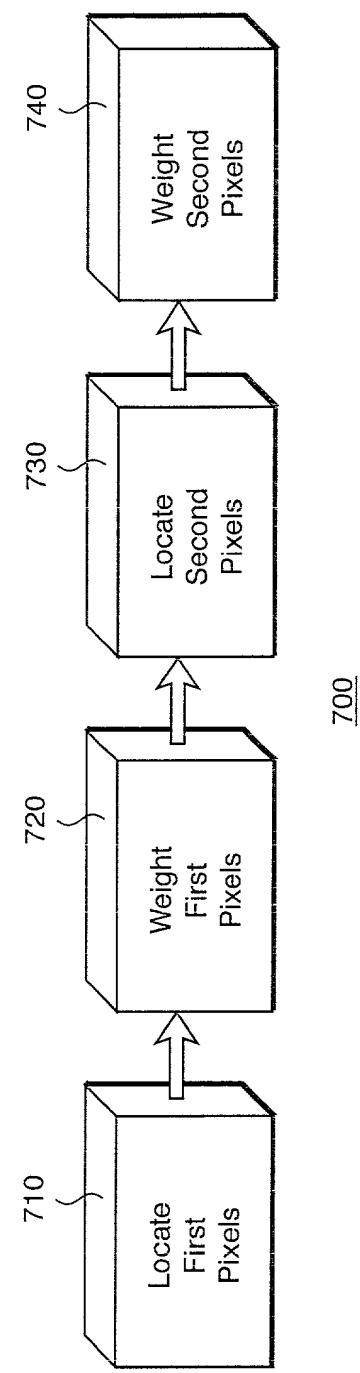


FIG. 7

```
At each projector,  
    Compute boundaries between regions of overlap count 1 and >1  
    Compute depth discontinuities using edge detection in depth buffer  
    For each pixel in overlap region  
        update shortest distance to overlap count = 1 region ignoring  
            paths crossing depth discontinuity  
  
At each projector,  
    For each pixel in overlap region  
        Find all corresponding pixels in other projectors  
        Assign weights inversely proportional to the shortest distance
```

800

FIG. 8

